

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-73216

(43)公開日 平成11年(1999)3月16日

(51)Int.Cl.⁶
G 0 5 B 19/418
15/02
19/18
19/4155
G 0 6 F 9/46 3 6 0

識別記号

F I
G 0 5 B 19/417 Q
G 0 6 F 9/46 3 6 0 B
G 0 5 B 15/02 W
19/18 U
V

審査請求 有 請求項の数1 O L (全 13 頁)

(21)出願番号 特願平10-165303
(62)分割の表示 特願平5-175490の分割
(22)出願日 平成5年(1993)7月15日

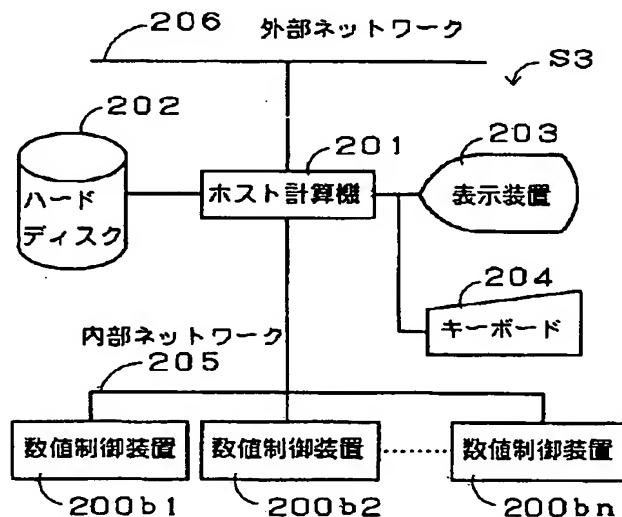
(71)出願人 000006013
三菱電機株式会社
東京都千代田区丸の内二丁目2番3号
(72)発明者 南角 茂樹
名古屋市東区矢田南五丁目1番14号 三菱
電機株式会社名古屋製作所内
(74)代理人 弁理士 宮田 金雄 (外2名)

(54)【発明の名称】 数値制御装置の制御ソフトウェア実行システム

(57)【要約】

【課題】 1プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムを用いたときにも、計算機と数値制御装置との間でタスクを分散して処理する。

【解決手段】 ホスト計算機201とそのホスト計算機201に内部ネットワーク205を介して接続された数値制御装置200bとの間に分散型共有メモリの機能を持たせ、その分散型共有メモリの機能を利用して、数値制御装置のタスクの機能を、前記ホスト計算機201と前記数値制御装置200bとで分散して処理する。



【特許請求の範囲】

【請求項1】 計算機とその計算機に通信回線を介して接続された数値制御装置との間に分散型共有メモリの機能を持たせ、リアルタイムオペレーティングシステム上で制御ソフトウェアを実行したときに起動される複数のタスクの共同作業によって数値制御装置としての機能を実現する数値制御装置のタスクの機能を、前記分散型共有メモリの機能を利用して、前記計算機と前記数値制御装置とで分散して処理することを特徴とする数値制御装置の制御ソフトウェア実行システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、数値制御装置の制御ソフトウェア実行システムに関し、さらに詳しくは、数値制御装置の制御ソフトウェア（コントロールプログラム）を効率的に実行できる数値制御装置の制御ソフトウェア実行システムに関する。

【0002】

【従来の技術】 図11に示す数値制御装置501のハードウェア構成図を用いて従来の数値制御装置の一般的構成を説明する。1は、各部の作動を制御したり、数値制御に必要な演算を行なったりするメインCPUである。2は、各部を結ぶシステムバスである。3は、数値制御装置の主要機能を実現する制御ソフトウェアなどを格納するROM（不揮発性の記憶装置）である。4は、一時記憶やワークエリアなどに用いるRAM（揮発性の記憶装置）である。5は、外部との間でシリアル通信によりデータのやり取りを行なうSIOインターフェース部である。6は、数値制御装置の運転状態を表示したり、数値制御装置に与えた指令を確認したりするための表示装置である。7は、数値制御装置に指令を与えるためのキーボード（入力装置）である。8は、サーボモータを制御するための指令を演算するサーボ制御部である。このサーボ制御部8は、メインCPU1とは別の専用CPUを備えてもよい。9は、サーボ制御部8から受け取った指令を増幅してサーボモータへ駆動信号を出力するサーボアンプである。10は、工作機械（図示せず）の加工部を制御するためのサーボモータである。11は、工作機械との間で、サーボ制御指令以外のデータをやり取りするためのプログラムコントローラ部である。12は、メインCPU1に与えられる、システムクロック（図示せず）と外部割り込み信号を表している。システムクロックは、数値制御装置全体を制御するためのクロック信号である。また、外部割り込み信号は、電源異常や非常停止などのイベント（緊急な出来事）の発生をメインCPU1に通知するための信号である。

【0003】 次に、上記の数値制御装置501の動作を説明する。メインCPU1は、ROM3に書き込まれている制御ソフトウェアをシステムバス2を通して1命令ずつ順に読み込んで実行する。図12に、命令読み込後の

2

処理手順を示す。加工プログラム入力処理21では、SIOインターフェース部5を通して外部から加工プログラム20を読み込み、RAM4に格納する。そして、加工プログラム20をブロック（所定の単位）ごとに内部データに変換する。補正計算処理22では、ブロックごとの内部データを処理し、増分移動量を算出する。また、工具径や工具長などを補正する。さらに、内部の座標値の更新処理を行なう。設定表示処理23では、数値制御装置の各種データを表示装置6に表示する。また、キーボード7を用いてオペレータが入力した各種設定データをRAM4に格納する。補間処理24では、補正計算処理22の処理結果を用いて、微小時間ごとの各軸の移動量を算出する。サーボ処理25では、補間処理24の処理結果を用いて、さらに小さな単位時間ごとの各軸の移動量に変換する。更に、サーボモータ10からのフィードバック制御（図示せず）を行なう。プログラムコントロール処理26では、工作機械との間での入出力処理や主軸の制御など、工作機械の周辺の制御などを行なう。

【0004】 さて、上記の数値制御装置501は、先に説明したようにメインCPU1に外部割り込み信号12を入力して割り込み処理を行なわせることで、非常停止などの緊急事態に対処することが出来る。メインCPU1は、外部割り込み信号12が入力されると、予め指定された別処理を実行し、それら別処理が終了した後に、通常の命令に復帰する。

【0005】 図13は、割り込み処理の概念図である。30～36は、通常の命令である。37～40は、割り込み命令である。41は、通常の命令への復帰命令である。例えば、メインCPU1が通常の命令33を実行中に外部割り込み信号12が入力されると、メインCPU1は、通常の命令33の処理終了後、割り込みを検出し、予め指定されていた割り込み命令37の実行を開始する。そして、割り込み命令37～40の実行を終了した後、復帰命令41を実行して、通常の命令34に復帰し、通常の命令35、36を実行する。なお、割り込み処理の処理時間が長くなると多重割り込みなどを考慮する必要が出てくるので、割り込み処理の処理時間をできるだけ短くすることが望ましい。

【0006】 ところで、上記の数値制御装置501の制御ソフトウェアは、以下に示すような特徴を有する。

①数値制御装置の制御ソフトウェアの機能ごとに、処理に必要な応答時間（ターンアラウンドタイム、デッドライン）が異なる。例えば、サーボ処理25（図28）は、処理結果の算出が遅れると切削が止って被加工物が不良品になってしまうので、実時間処理でなくてはならない。一方、表示装置6への表示処理などは、多少遅れても不都合は生じないので、実時間処理でなくてもよい。

②メインCPU1が実行すべき割り込み処理に、多くの種類がある。これらの理由により、数値制御装置501

3

の制御ソフトウェアは、一般にリアルタイムオペレーティングシステムの制御のもとで実行される機能ごとのタスクを実行単位とすることが多い。

【0007】次に、リアルタイムオペレーティングシステムにより各タスクを制御する方法を説明する。あるタスクを定期的に実行させたり、あるタスクを実行する時間制限したりするために、通常はメインCPU1に対してある一定周期で内部割り込みを入れる。(これをシステムクロックと呼ぶ)

リアルタイムオペレーティングシステムは、システムクロック割り込みがある度に各タスクの状態を調べ、実行中のタスクを止めて別のタスクを実行させたりする。
(これをスケジューリング、またはディスパッチと呼ぶ)

また、リアルタイムオペレーティングシステムにおいては、各タスクに優先順位(実行の優先度)づけを行なう。この優先順位の意味は、より低い優先順位のタスクを実行中に、より高い優先順位のタスクを実行する準備ができたときは、その優先順位の低いタスクの実行を中断させて優先順位の高いタスクを実行させることをいう。(これを横取りといふ)

【0008】図14は、各タスク動作の時間的関係を示すタイムチャートである。縦軸は各処理の実行状況(実行または停止)を表し、横軸は時間の経過を表す。なお、図面作成の都合上、各処理としては、リアルタイムオペレーティングシステムと、割り込み処理と、(優先順位が高い順に)サーボ処理タスク、補正計算処理タスク、表示設定処理タスクを用いるものとする。P1は、リアルタイムオペレーティングシステムが動作している最中に割り込みが発生して制御が割り込み処理に移ったことを示す。P2は、割り込み処理が終了した後、リアルタイムオペレーティングシステムのスケジューラに制御が戻る様子を示す。なお、スケジューラは、リアルタイムオペレーティングシステムでスケジューリングを行なう部分である。P3は、リアルタイムオペレーティングシステムのスケジューラにより、次に実行するタスクとしてサーボ処理タスクが選ばれたことを示す。P4は、サーボ処理タスクが終了した後、リアルタイムオペレーティングシステムに制御が戻る様子を示す。P5は、リアルタイムオペレーティングシステムのスケジューリングによって、次に実行するタスクとして補正計算処理タスクが選ばれたことを示す。P6は、補正計算処理タスクが終了した後、リアルタイムオペレーティングシステムに制御が戻る様子を示す。P7は、より優先順位の高いタスクを実行する準備ができていないので、優先順位が最も低い表示処理設定処理タスクに移ったことを示す。P8は、割り込みが発生したことで表示処理設定処理タスクを中断し、割り込み処理に制御が移ったことを示す。P9は、割り込み処理を終了した後、リアルタイムオペレーティングシステムのスケジューラに制御

4

が戻る様子を示す。(一般に、割り込み処理を終了した後は、もとのタスクに制御は戻らず、オペレーティングシステムのスケジューラに制御が戻る。)

P10は、再び表示設定処理タスクに制御が移ったことを示す。P11は、システムクロック(図示せず)によってリアルタイムオペレーティングシステムのスケジューラに制御が戻ったことを示す。P12は、サーボ処理タスクを実行する準備ができたので、サーボ処理タスクに移ったことを示す。(これは、表示設定タスクがサーボ処理タスクに実行権を横取りされた例である)

P13は、サーボ処理タスクはその実行を終了した後、自動的に実行権を放棄して、リアルタイムオペレーティングシステムのスケジューラに制御を戻したことを示す。P14は、再び表示設定処理タスクに制御が移ったことを示す。以降、各タスクは同様の動作を繰り返して、処理を続ける。

【0009】なお、図14に示したように、システムクロック以外の割り込みは不定期に発生するため、数値制御装置側ではその発生を予想できない。そこで、各タスクの特性に応じて、処理時間の配分を行なう必要がある。

例えれば、サーボ処理タスクは、前もって所定の単位時間のサーボモータの一回分の移動量を計算しておくために、所定の単位時間以内に次の移動量を計算しておく必要がある。つまり、少なくとも所定の単位時間以内には繰り返して実行する必要がある。一方、補正計算処理タスクは、例えば1回の実行で、サーボ処理タスクを3回だけ実行するために必要なデータを作成できるので、単位時間3回につき1回だけ実行すれば良い。また、表示設定処理タスクは他の処理がない時だけ実行すれば足る。

【0010】以上で述べたように、数値制御装置501は、リアルタイムオペレーティングシステム上で制御ソフトウェアを実行したときに起動される複数のタスクの共同作業によって数値制御装置としての機能を実現する。

【0011】さて、現在、数値制御装置の制御ソフトウェアの開発マシンとしては、ワークステーションが主流である。そこで、図15に、ワークステーションを用いた数値制御装置の制御ソフトウェア開発のシステム構成を例示する。100は、コンパイラ、数値制御装置の制御ソフトウェアのソースコードやオブジェクトコードなど、各種ファイルを格納しているハードディスクである。101は、各種ファイルを一括管理したり、クライアントワークステーションの要求に応じてファイルを転送したりするファイルサーバである。102は、他のネットワークと通信するためのゲートウェイマシンである。103は、各種I/Oデバイスを制御するためのI/Oサーバである。104は、他のシステムに繋がっているネットワークである。105は、システム内を結ぶローカルネットワークである。106~108は、クラ

10

20

30

40

50

イアントワークステーションである。

【0012】ワークステーションおよびワークステーションのオペレーティングシステムについて説明する。ワークステーションには、以下に示すような特徴がある。
①ワークステーションにおいては多くのユーザが同時に使用して、その各ユーザに対して公平なオペレーティングシステムのサービスが要求される。つまり、ユーザによる優先順位のようなものも普通は存在しないし、また緊急の割り込みというものもほとんど存在しない。

②必ず決まった時間以内に処理しなくてはならない処理は、ほとんど無い。そのため、ワークステーションのオペレーティングシステムとしては、多数の利用者が端末と対話しながら処理を行えるタイムシェアリングオペレーティングシステムを用いることが多い。

【0013】タイムシェアリングオペレーティングシステムにおいて、あるまとまった処理の単位はプロセスと呼ばれる。プロセスは、リアルタイムオペレーティングシステムにおけるタスクと類似している。但し、一般に、リアルタイムオペレーティングシステムのもとでの各タスクのアドレス空間は1つの同じものである（線形アドレス空間と呼ぶ）のに対して、タイムシェアリングオペレーティングシステムでは、プロセスごとに固有なアドレス空間を持っている。具体例で示せば、リアルタイムオペレーティングシステムでは、CPUがアクセスして出来るアドレス（これを論理アドレス空間と呼ぶ）が0番地から100000番地まであったと想定すると、リアルタイムオペレーティングシステムが0番地から10000番地を占め、タスク1が10001番地から20000番地を占め、タスク2が20001番地から28000番地までを占める。タスク1の20000番地もタスク2の20000番地と同じところを指し、同じデータを示す。そこで、タスク間でアドレスをやり取りすることで、お互いのデータなどを参照できる。一方、タイムシェアリングオペレーティングシステムでは、プロセスごとに固有なアドレス空間を持っているので、プロセス1とプロセス2がどちらもアドレス0番地からアドレス100000番地までを持っていると想定すると、プロセス1の20000番地とプロセス2の20000番地では実際のメモリ上の異なるところを指すので、そのデータも異なる。

【0014】次に、プロセスごとに固有なアドレス空間の実現方法を説明する。図16は、クライアントワークステーションのメモリ構造図である。130はクライアントワークステーション、131はCPU、132はアドレス変換機構、133は変換テーブル、134は主記憶（メモリ）、135はネットワーク、136はファイルサーバ、137はハードディスクである。

【0015】さて、クライアントワークステーション130において、CPU131で区別できるアドレス空間（論理アドレス空間）は、一般に、主記憶につけられた

物理的な番地（物理アドレス空間）よりも大きい。そこで、仮想記憶の手法により、物理アドレス空間外の論理アドレスをハードディスクなどの2次記憶装置に割り当てることで、2次記憶装置をも主記憶とみなせるようになる。つまり、主記憶の大きさが実装容量に制限されなくなるので、アドレス空間の制限を気にせずにソフトウェアを作成できるようになる。

【0016】次に、仮想記憶の手法をさらに詳しく説明する。図16において、CPU131はアドレス変換機構132に対して論理アドレスを送る。アドレス変換機構132では、与えられた論理アドレスの一部または全部を取り出して変換テーブル133の配列の何番目かを示すインデックスとして用いる。変換テーブル133の内容が主記憶134のアドレスを示していれば主記憶134に対して所定の動作を行なう。また、変換テーブル133の内容がハードディスク137のブロック番号などを示していればファイルサーバ136に対してブロック番号に対応した内容を要求し、ネットワーク135を介して内容を受け取る。主記憶134に未使用領域があれば、受け取ったデータを未使用領域に格納し、データを格納した領域を示すように変換テーブル133の内容を書き換える。また、主記憶134に未使用領域が存在しなければ、主記憶134に未使用領域が無ければ、主記憶134の既使用領域を選び、その内容をハードディスク137に書き戻し、その領域を新たに使用する。なお、主記憶134から変換テーブル133を逆にたどるソフトウェアの機能も備えられている。（これをコアマップと呼ぶ）

【0017】さらに、プロセスごとに変換テーブルを持つことにより、各プロセスが同じ論理アドレス空間を利用することが出来る。この様子を図17に示す。140はプロセス1、141はプロセス2、142はプロセス1の変換テーブル、143はプロセス2の変換テーブル、144は実メモリを示す。いま、プロセス1（140）とプロセス2（141）のいずれも、アドレス0番地（論理アドレスの0番地）をアクセスする場合を想定する。このとき、プロセス1（140）は、変換テーブル142の0番地の部分を指している。また、プロセス2（141）は、変換テーブル143の0番地の部分を指している。ところで、変換テーブル142の0番地と変換テーブル143の0番地は、実メモリ144上の異なる領域を指している。つまり、各プロセスが論理アドレス空間をフルに利用することが出来る。

【0018】各プロセスは、例えば図18に示すように、論理アドレス空間をいくつかの領域に分けて使用する。155は、1つのプロセスのすべての論理アドレス空間を示す。156は、そのプロセスのテキスト（プログラム）領域を示す。158は、予備領域である。この予備領域は、普段は未使用領域であるが、データ領域やスタック領域が不足したときには、図示の矢印の方向

で、データ領域やスタック領域として用いられる。15
7は、そのプロセスが使用するデータ領域を示す。15
9は、スタック領域を示す。

【0019】図19は、タイムシェアリングオペレーティングシステムにおけるプロセスモデルを示す。210は計算機、211はプロセス、212は実体（例えばテキストコード）、213は実体において現在実行中のコードを示すプログラムカウンタである。各プロセスは、アドレス空間や、そのプログラムカウンタや、スタックや、レジスタ群などを持っている。なお複数のプロセス間で協調して処理を行なうときには、プロセス間通信の手段を用いて相互に連絡をとる必要がある。

【0020】上記のタイムシェアリングオペレーティングシステムでは、各プロセスが固有のアドレス空間を持つことで、プロセスごとの変換テーブルの書き換えなどに手間がかかり、プロセスの切り換えに時間がかかりやすい欠点がある。また、各プロセス内の実行を行なう実体は1個であるので、複雑な処理を行なうアプリケーションプログラムに対応しづらくなってきた。そのため、近年、1つのプロセス中に複数の実体を置き、各実体を共通のアドレス空間のもとで別々のプロセスであるかのように動かせるマルチスレッドオペレーティングシステムが提案されている。

【0021】図20に、マルチスレッドオペレーティングシステムのモデルを示す。210は計算機、211はプロセス、212は実体、213はプログラムカウンタである。実体212aを、スレッド（またはライトウェイトプロセス）と呼ぶ。このスレッドは、プロセスと同様に、固有のプログラムカウンタやスタックを備えており、他のプロセスとは独立して動く。1つのプロセス内の全てのスレッドは同じアドレス空間を持っているので、共通変数を共有することも容易である。つまり、先に説明したタイムシェアリングオペレーティングシステムのプロセスモデルでは、各プロセスが1つずつしかスレッドを持てないのでに対して、マルチスレッドオペレーティングシステムのプロセスモデルでは、1つのプロセスが複数のスレッドを持つことができる。

【0022】一般的に、各プロセスが持っていないなければならない情報には次のようなものがある。

- ・アドレス空間
- ・共通変数（大域変数）
- ・開いているファイルの情報
- ・自分が生成したプロセスの情報
- ・タイマー情報
- ・シグナル情報
- ・セマフォーの情報

一方、スレッド毎に持つべき情報には次のようなものがある。

- ・プログラムカウンタ
- ・スタック

- ・レジスタセット
- ・自分が生成したスレッドの情報
- ・自分自身の状態

【0023】1つのプロセス中のすべてのスレッドは、プロセスごとに持つべき上記の情報を共有している。一方、スレッドごとに持つべき上記の情報は、各スレッドが個々に持っている。例えば図21に示すように、1つのプロセスのすべての論理アドレス空間155は、テキスト領域156と、データ領域157と、予備領域158と、スレッド1のスタック領域220と、スレッド2のスタック領域221と、スレッド3のスタック領域222と、スレッド「N-1」のスタック領域223と、スレッドNのスタック領域224とに分かれている。つまり、1つのプロセス内の各スレッドが個別に持たねばならない領域のみを別個に割り当てるこによって、1つのプロセス内に複数のスレッドを構成できる。

【0024】次に、マルチスレッドオペレーティングシステムにおけるスケジューリングを説明する。図14を用いて先に説明したリアルタイムオペレーティングシステムではタスクごとにスケジューリング行なったのに対して、マルチスレッドオペレーティングシステムでは、スレッドごとにスケジューリングを行なう。但し、マルチCPUのときにはCPUごとにスレッドが割り当てられるので、割り当てられたスレッドごとのスケジューリングを行なえばよい。

【0025】

【発明が解決しようとする課題】上記従来の数値制御装置501では、数値制御装置のメインCPUとして、高性能なCPUを使用すると、CPU性能をフルに發揮していない間の余力が無駄となり、コスト高となる。一方、低性能なCPUを使用すると、低コストとなる代りに、数値制御装置としての機能に支障が出たり、処理時間が長くなったりする。この発明は、上記のような問題点を解決するためになされたもので、低成本で効率的に数値制御装置の制御ソフトウェアを実行できる数値制御装置の制御ソフトウェア実行システムを得ることを目的とする。

【0026】

【課題を解決するための手段】この発明は、計算機とその計算機に通信回線を介して接続された数値制御装置との間に分散型共有メモリの機能を持たせ、リアルタイムオペレーティングシステム上で制御ソフトウェアを実行したときに起動される複数のタスクの共同作業によって数値制御装置としての機能を実現する数値制御装置のタスクの機能を、前記分散型共有メモリの機能を利用して、前記計算機と前記数値制御装置とで分散して処理することを特徴とする数値制御装置の制御ソフトウェア実行システムを提供する。上記数値制御装置の制御ソフトウェア実行システムでは、計算機と数値制御装置との間でタスクを分散して処理するので、数値制御装置には低

性能のCPUを使用でき、低コストにして効率的に数値制御装置の制御ソフトウェアを実行することが出来る。さらに、分散型共有メモリの機能を利用するため、1プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムを用いたときにも、計算機と数値制御装置との間でタスクを分散して処理することが出来る。

【0027】

【発明の実施の形態】図1は、この発明の一実施形態の数値制御装置の制御ソフトウェア実行システムの全体構成図である。この制御ソフトウェア実行システムS3は、数値制御装置200b1～200bnと、ホスト計算機201と、ハードディスク202と、表示装置203と、キーボード204とを備える。数値制御装置200b1～200bnとホスト計算機201とは、内部ネットワーク205を介して接続されている。数値制御装置200b1～200bnは、リアルタイムオペレーティングシステムのもとで制御ソフトウェアを実行して、数値制御装置としての機能を実現する。ホスト計算機201は、汎用のマルチスレッドオペレーティングシステムのもとで数値制御装置の制御ソフトウェアを実行する。ハードディスク202は、ホスト計算機201の外部記憶装置である。表示装置203は、ユーザが必要とする各種の情報を表示する。キーボード204は、ユーザからの指示などを受け付ける。外部ネットワーク206は、ホスト計算機201を、汎用のワークステーション(図示せず)などと結ぶ。

【0028】図2は、数値制御装置200b1のハードウェア構成図である。なお、数値制御装置200b2～200bnも同じ構成である。1は、各部の作動を制御したり、数値制御に必要な演算を行なったりするメインCPUである。2は、各部を結ぶシステムバスである。3は、数値制御装置の主要機能を実現する制御ソフトウェアなどを格納するROM(不揮発性の記憶装置)である。4は、一時記憶やワークエリアなどに用いるRAM(揮発性の記憶装置)である。5は、外部との間でシリアル通信によりデータのやり取りを行なうデータSI/Oインターフェース部である。6は、数値制御装置の運転状態を表示したり、数値制御装置に与えた指令を確認したりするための表示装置である。7は、数値制御装置に指令を与えるためのキーボード(入力装置)である。8は、サーボモータを制御するための指令を演算するサーボ制御部である。9は、サーボ制御部8から受け取った指令を増幅してサーボモータへ駆動信号を出力するサーボアンプである。10は、工作機械(図示せず)の加工部を制御するためのサーボモータである。11は、工作機械との間で、サーボ制御指令以外のデータをやり取りするためのプログラムコントローラ部である。12は、メインCPU1に与えられる、システムクロック(図示せず)と外部割り込み信号を表している。システムクロ

ックは、数値制御装置全体を制御するためのクロック信号である。また、外部割り込み信号は、電源異常や非常停止などのイベント(緊急な出来事)の発生をメインCPU1に通知するための信号である。132は、図16を用いて先に説明した仮想記憶を実現するためのアドレス変換機構である。207は、内部ネットワーク205を通じてホスト計算機201や他の数値制御装置200b2～200bnとの間で通信を行なうためのネットワークインターフェースである。

【0029】次に、ホスト計算機201により、数値制御装置の制御ソフトウェアを実行する手順を説明する。なお、説明の都合上、数値制御装置の制御ソフトウェアは、予めハードディスク202に格納されているものとする。まず、ユーザは、キーボード204を用いて数値制御装置の制御ソフトウェアを実現するプロセスの起動をホスト計算機201に指示する。ホスト計算機201のマルチスレッドオペレーティングシステムは、数値制御装置の制御ソフトウェアを実現するためのプロセスをハードディスク202から読み出し、プロセス管理テーブルに登録する。

【0030】プロセス管理テーブルは、概念的には、図16と図17を用いて先に説明した変換テーブル133と類似している。具体例で説明すれば、プロセス管理テーブルは、変換テーブル133の配列の1番目からM番目までを図21に示したテキスト領域156を示すためのスロットとして確保し、変換テーブル133の配列の「M+1」番目からN番目までを図21に示したデータ領域157を示すためのスロットとして確保したものである。

【0031】次に、ホスト計算機201は、数値制御装置の制御ソフトウェアの実行状況に伴って、図21に示したプロセスの論理アドレス空間155内の各領域を順にハードディスク202から読み出して内部メモリ(図示せず)へロードイングする。

【0032】次に、ホスト計算機201は、数値制御装置の制御ソフトウェアを実行したときのタスクに相当するプロセスを生成する。図3に、プロセスを生成するための流れ図を示す。まず、ステップ241では、マルチスレッドオペレーティングシステムに備えられているシ

40ステムコールを利用して、数値制御装置としての1機能を実現している1タスク(数値制御装置のリアルタイムオペレーティングシステムのもとでの1タスク)と同じ処理を行なう1スレッドを生成する。具体的には、プロセス管理テーブルに新たなスレッドのスタック領域用のスロットを確保し、マルチスレッドオペレーティングシステムが認識するスレッドの表の中に新たなスレッドを追加する。

【0033】ステップ242では、ステップ241で生成したスレッドに対応した処理を行なう。例えば、数値制御装置のリアルタイムオペレーティングシステムに対

11

するシステムコールを、ホスト計算機201のマルチタスクオペレーティングシステムのスレッドに対するシステムコールに書き換える。なお、変換ライブラリを用いてシステムコールを変換するならば、書き換え負担を軽減できる。

【0034】ステップ243では、数値制御装置のタスクに相当するスレッドをすべて生成したか否か判定する。すべてのスレッドを生成したならばステップ244に進み、生成すべきスレッドが残っていればステップ241に戻る。なお、すべてのスレッドを生成した時点では、ホスト計算機201の内部には、図4の概念図に示すように、マルチスレッドオペレーティングシステム240上に、数値制御装置のタスクと同じ機能を実現するスレッド1(231)～スレッド6(236)を含むプロセス211が構成されることになる。

【0035】ステップ244では、スレッド間のスケジューリングを行なうスケジューリング方式を決定する。例えば、優先順位に応じて各スレッドを走らせるスケジューリング機能をホスト計算機201のマルチスレッドオペレーティングシステムが備えている場合には、システムコールを用いて各スレッドの優先順位を登録する。又は、ホスト計算機201のマルチスレッドオペレーティングシステムは、準備ができた順序でスレッドを走らせるスケジューラを有していることもある。ここで、“走る”とは、そのスレッドを構成するテキスト領域が実メモリ上に存在し、そのテキスト領域をCPUが処理している状態のことである。

【0036】図5は、スケジューラによるスケジューリングの一例を示す概念図である。211は、ホスト計算機201が実行するプロセスである。231～234は、数値制御装置のタスクと同じ機能を実現するスレッドである。240は、マルチスレッドオペレーティングシステムである。250は、準備が出来たスレッドの中から1つのスレッドを選び順に走らせるスケジューラである。251は、他のプロセスである。

【0037】ステップ245では、数値制御装置の制御ソフトウェアが最初に初期化タスクを起動し、その初期化タスクが他のタスクを起動するような方式であるときは、初期化タスクと同じ処理を行なうスレッドをシステムコールにより起動する。具体的には、マルチスレッドオペレーティングシステムのスケジューリング対象のスレッドの表の中に、初期化タスクと同じ処理を行なうスレッドを加えればよい。これらの処理により、各スレッドは、予め決定された順序に応じて走ることになる。なお、数値制御装置の制御ソフトウェアが上記のような方式でない場合には、従来の数値制御装置の制御ソフトウェアと同じ順序で該当スレッドを起動すればよい。

【0038】上記ではホスト計算機201のマルチスレッドオペレーティングシステムの機能を利用してスレッド間のスケジューリングを行なったが、スレッド間のス

10

ケジューリングの制御用に1つのスレッドを生成するようにもよい。つまり、ホスト計算機201は、数値制御装置における各タスクをスレッドとするのに加えて、スケジューラ250と同じ働きをするスケジューラスレッドを生成する。

【0039】図6は、ホスト計算機201により生成されたスケジューラスレッドによるスケジューリングの概念図である。これは、図5を用いて先に説明したスケジューリングの概念に、スケジューラスレッド252を加えたものである。そして、スケジューラスレッド252は、マルチスレッドオペレーティングシステム240上のスケジューラ250から数値制御装置の制御ソフトウェアを実行するプロセス211の実行権を獲得し、スレッド1(231)～スレッド4(234)のスケジューリングをすべて行なう。

20

【0040】数値制御装置の制御ソフトウェアのタスクに相当するスレッドは、ホスト計算機201と数値制御装置200b1との間でスレッド間通信を行なうこと、分散して処理される。スレッドの分散の具体例を説明すれば、ある数値制御装置200b1が低性能で安価なメインCPUを有しており、ホスト計算機201が高性能で高価なCPUを有しているときには、簡単な処理を行なうスレッドは数値制御装置200b1に走らせて、複雑な処理を行なうスレッドはホスト計算機201に走らせるのが好適である。又、別の具体例を説明すれば、図12を用いて先に説明した補間処理24やサーボ処理25を行なうスレッドなどは数値制御装置200b1に走らせ、加工プログラム入力処理21や補正計算処理22を行なうスレッドはホスト計算機201に走らせるのが好適である。

30

【0041】図7に示す概念図を用いて、ホスト計算機201と数値制御装置200b1との間で行なわれるスレッド間通信について説明する。211は、プロセスである。231はスレッド1、232はスレッド2、233はスレッド3、234はスレッド4である。260はノード1、261はノード2である。ここで、ノードとは、ネットワークの枝の端点にあたる情報処理機器の総称である。例えば、ノード1(260)はホスト計算機201、ノード2(261)は数値制御装置200b1である。262は、オペレーティングシステム240aのもとでスレッド1とスレッド2との間の通信を処理するスレッド間通信処理部である。263は、ノード1とノード2とを結ぶノード間ネットワークである。

40

【0042】いま、ノード1(260)内のスレッド1(231)からノード2(261)内のスレッド3(233)にメッセージを送る場合を想定する。スレッド1(231)はシステムコールによって相手スレッドを指定し、メッセージを送る要求をオペレーティングシステム240aに対して発行する。このシステムコールによって制御は、オペレーティングシステム240aに移

50

る。システムコールの内容を解析することで、制御はスレッド間通信処理部262に移る。スレッド間通信処理部262は、相手スレッドがあるノード（ノード2）を判定し、ノード2（261）に対してノード間ネットワーク263を介してメッセージを送る。そのメッセージは、ノード2（261）のスレッド間通信処理部262に渡り、スレッド間通信処理部262は、指定されたスレッド3（233）に対してメッセージを渡す。

【0043】上記の想定では、スレッド1（231）からスレッド3（233）に直接にメッセージを送るものとしたが、マルチスレッドオペレーティングシステムによってはメールボックスを用いてメッセージを渡すことも出来る。図8は、ノード1（260）内のスレッドがノード2（261）内のスレッド3（233）にメールボックスを用いてメッセージを送る場合の概念図である。スレッド1（231）はシステムコールによって相手スレッドのメールボックスを指定し、メッセージを送る要求をオペレーティングシステム240aに対して発行する。このシステムコールによって、制御はオペレーティングシステム240aに移る。システムコールの内容を解析することで、制御はスレッド間通信処理部262に移る。スレッド間通信処理部262は指定されたメールボックスがあるノード（ノード2）を判定して、ノード2（261）に対してノード間ネットワーク263を通してメッセージを送る。そのメッセージはノード2（261）のスレッド間通信処理部262に渡る。スレッド間通信処理部262は、メールボックス264にメッセージを格納する。その後、スレッド間通信処理部262は、スレッド3（233）からポートを指定したメッセージ読み込み要求があったらメッセージを渡す。読み込み要求の方がメッセージの到着よりも速かった場合には、メッセージが到着するまでスレッド3（233）を待機させる。

【0044】1プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムでは、分散型共有メモリの機能を利用して、計算機201と数値制御装置200b1とで分散して処理する。すなわち、ホスト計算機201に数値制御装置の制御ソフトウェアの機能を実現するプロセスを走らせ、1つの数値制御装置200b1にホスト計算機201のプロセスとは独立したスレッドを走らせる。その際、1プロセス内のスレッドでのアドレス空間の共有を、分散型共有メモリの機能により実現する。

【0045】図9は、分散型共有メモリの概念図である。ノード1（260）のCPU131が、ある論理アドレスに対応した主記憶134の内容を読むときは、アドレス変換機構132と変換テーブル133を用いて、主記憶134の物理アドレスを算出して、内容を得る。例えば、論理アドレスがノード1（260）の主記憶134に対応しているならば、ノード1（260）の変換

テーブル133は主記憶134の物理アドレスを算出する。また、論理アドレスがノード2（261）の主記憶134に対応しているならば、ノード1（260）の変換テーブル133は、ノード間ネットワーク135を介してノード2（261）の変換テーブル133に主記憶134の内容の読み出しを要求する。

【0046】上記では、ホスト計算機201と1つの数値制御装置200b1とに分散してスレッドを走らせたのに対して、図10に示すように、ホスト計算機201と複数の数値制御装置200b1～200bnとに分散してスレッドを走らせてよい。例えば、実時間処理が必要な補間処理24やサーボ処理25（図12）を行なうスレッド231は、それぞれの数値制御装置200b1～200bnで走らせ、実時間処理が必要な加工プログラム入力処理21や補正計算処理22（図12）を行なうスレッド232は、ホスト計算機201上で共通に走らせるのが好適である。なお、ホスト計算機201上のスレッド232は、数値制御装置200b1～200bn対応で少なくとも1つずつ走らせる。

【0047】以上の数値制御装置の制御ソフトウェア実行システムS3によれば、1プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムを用いたときにも、計算機201と数値制御装置200bとの間でタスクを分散して処理するので、低コストにして効率的に数値制御装置の制御ソフトウェアを実行することが出来る。

【0048】

【発明の効果】この発明の数値制御装置の制御ソフトウェア実行システムによれば、計算機と数値制御装置との間でタスクを分散して処理するので、数値制御装置には低性能のCPUを使用でき、低コストにして効率的に数値制御装置の制御ソフトウェアを実行することが出来る。さらに、分散型共有メモリの機能を利用するため、1プロセス内の別スレッドを別ノードで分散して走らせることができないマルチスレッドオペレーティングシステムを用いたときにも、計算機と数値制御装置との間でタスクを分散して処理することが出来る。

【図面の簡単な説明】

【図1】この発明の一実施形態の数値制御装置の制御ソフトウェア実行システムを示す全体構成図である。

【図2】数値制御装置のハードウェア構成図である。

【図3】プロセスを生成するための流れ図である。

【図4】ホスト計算機の内部に構成されたプロセスの概念図である。

【図5】スケジューラによるスケジューリングの概念図である。

【図6】スケジューラスレッドによるスケジューリングの概念図である。

【図7】数値制御装置とホスト計算機の間で行なわれるスレッド間通信の概念図である。

15

【図8】メールボックスを介したスレッド間通信の概念図である。

【図9】分散型共有メモリの概念図である。

【図10】ホスト計算機と複数の数値制御装置とに分散してスレッドを走らせる説明図である。

【図11】従来の数値制御装置のハードウェア構成図である。

【図12】従来の数値制御装置の処理手順を示す説明図である。

【図13】割り込み処理の概念図である。

【図14】各タスク動作の時間的関係を示すタイムチャートである。

【図15】ワークステーションを利用した制御ソフトウェア開発システムの構成図である。

【図16】クライアントワークステーションのメモリ構造図である。

【図17】各プロセスが同じ論理アドレス空間を利用できることを示す原理図である。

【図18】1つのプロセスの論理アドレス空間内の領域図である。

【図19】タイムシェアリングオペレーティングシステムにおけるプロセスモデルの例示図である。

【図20】マルチスレッドオペレーティングシステムにおけるモデルの例示図である。

【図21】1つのプロセスの論理アドレス空間内の別の*

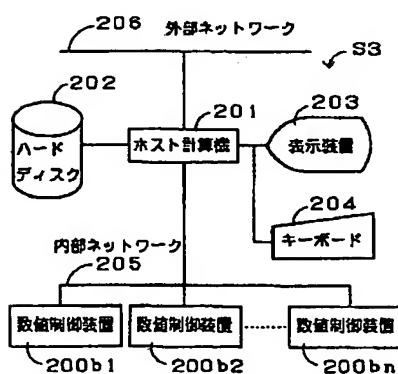
16

*領域図である。

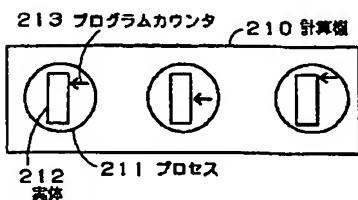
【符号の説明】

1	メインC P U
2	システムバス
3	R O M
4	R A M
5	S I O インタフェース部
6	表示装置
7	キーボード
8	サーボ制御部
9	サーボアンプ
10	サーボモータ
11	プログラムコントローラ部
12	外部割り込み信号
13	アドレス変換機構
S 3	制御ソフトウェア実行システム
2 0 0 b 1, …, 2 0 0 b n	数値制御装置
2 0 1	ホスト計算機
2 0 2	ハードディスク
2 0 3	表示装置
2 0 4	キーボード
2 0 5	内部ネットワーク
2 0 6	外部ネットワーク
2 0 7	ネットワークインターフェース
3	ROM
4	RAM
5	S I O インタフェース部
6	表示装置
7	キーボード
8	サーボ制御部
9	アンプ
10	モータ
11	プログラムコントローラ部
12	外部割り込み信号
13	アドレス変換機構
14	内部ネットワーク
15	外部ネットワーク
16	データ領域
17	予備領域
18	スタック領域

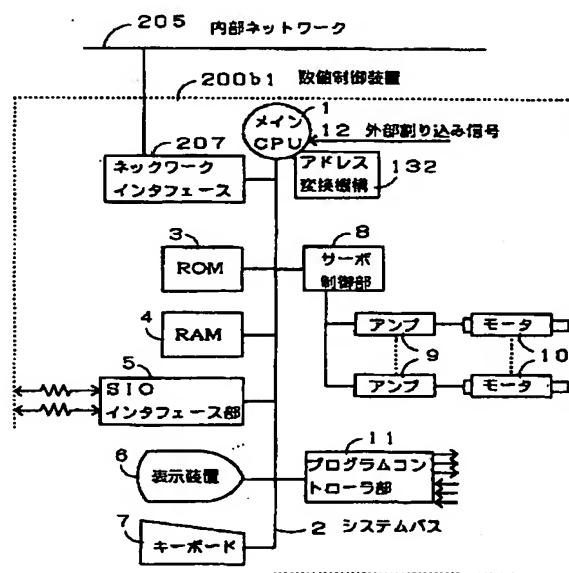
【図1】



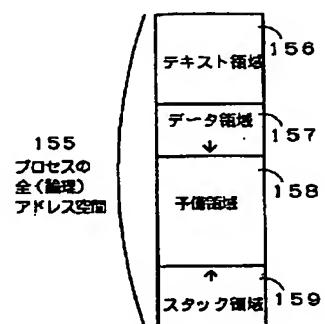
【図19】



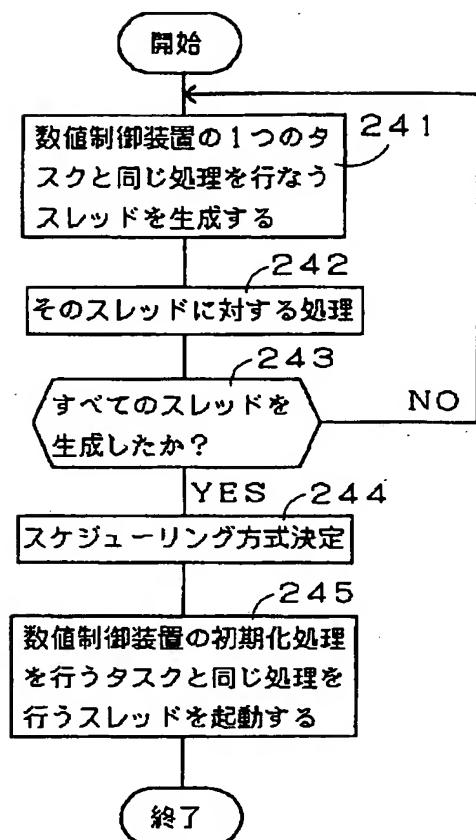
【図2】



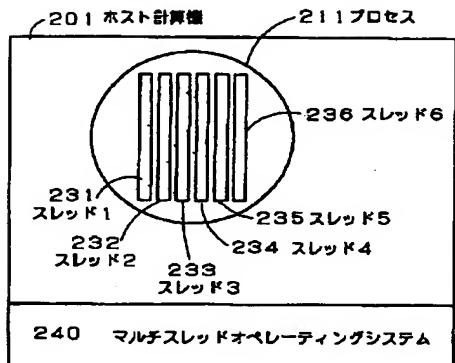
【図18】



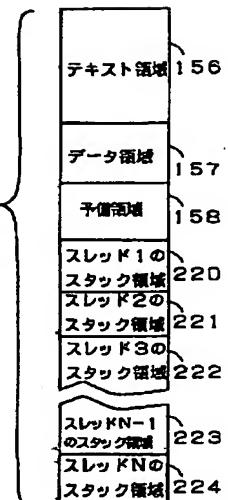
【図3】



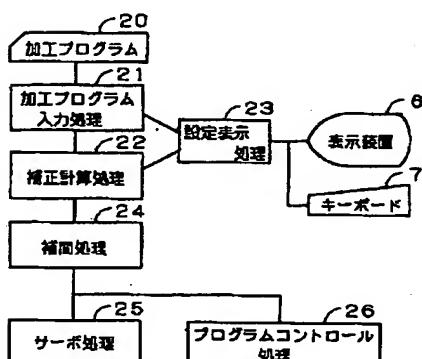
【図4】



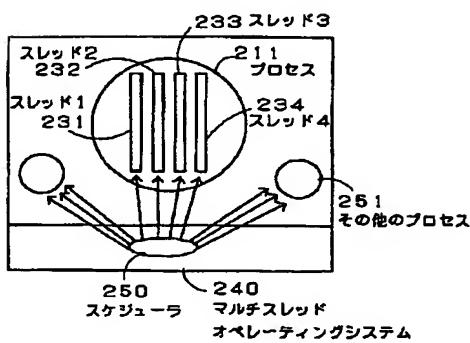
【図21】



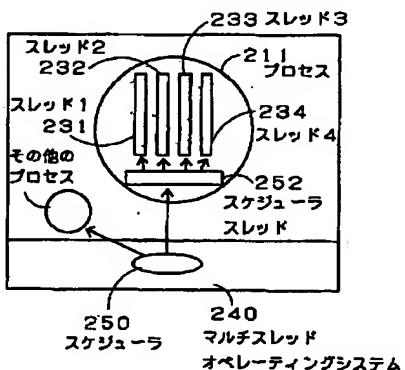
【図12】



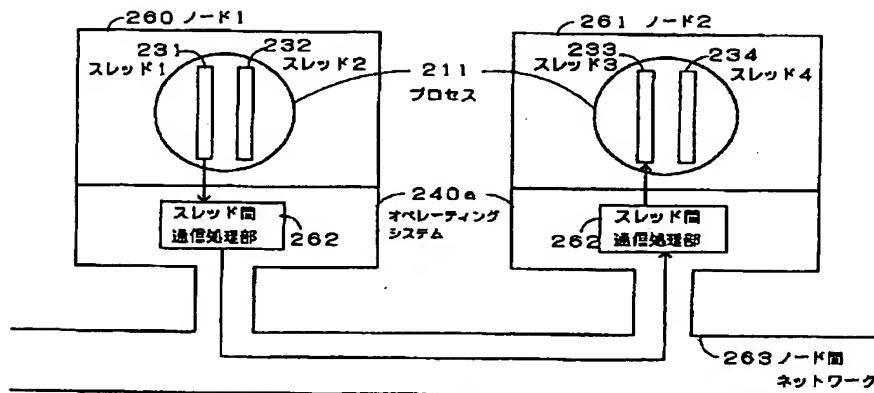
【図5】



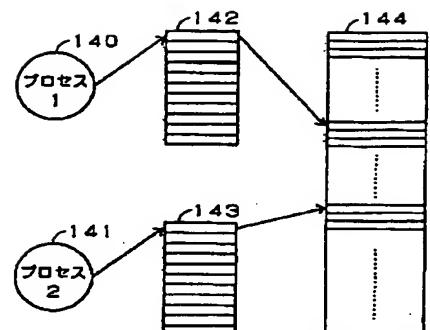
【図6】



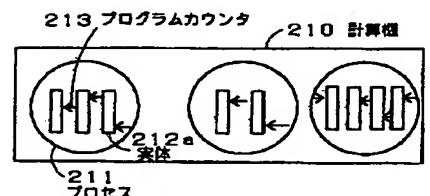
【図7】



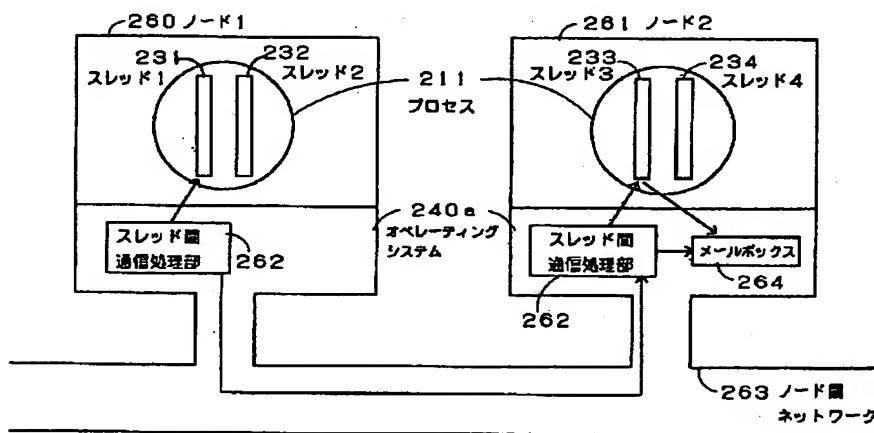
【図17】



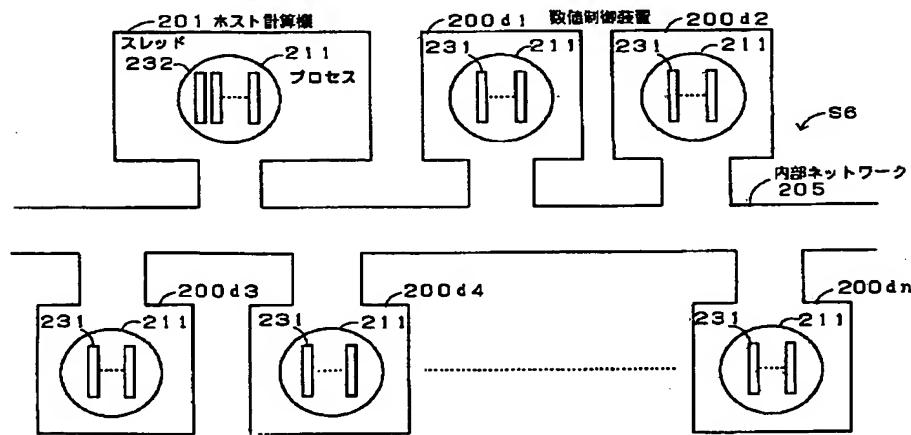
【図20】



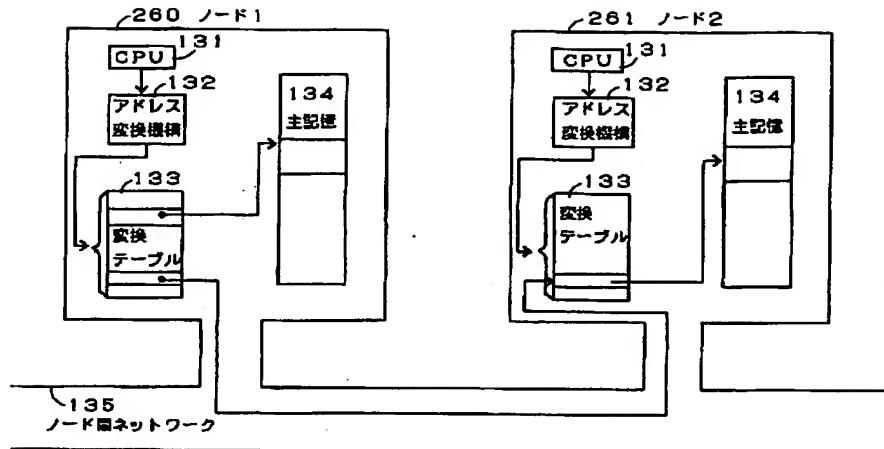
【図8】



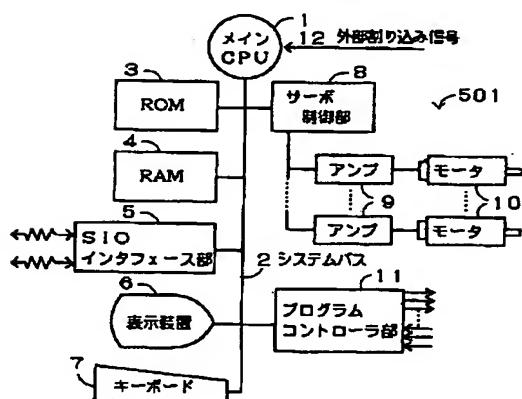
【図10】



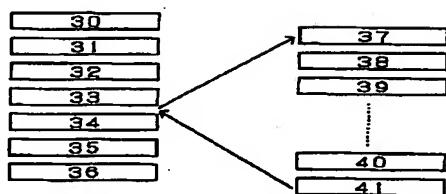
【図9】



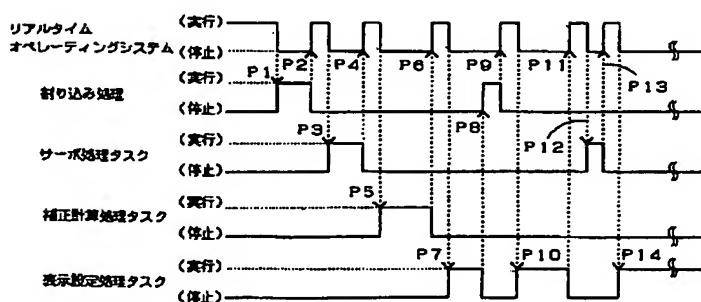
【図11】



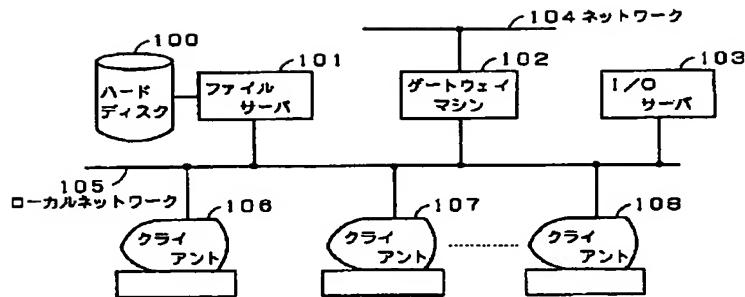
【図13】



【図14】



【図15】



【図16】

